CS422 Data Mining Project Report: Technical Report on Second-hand Housing Price Prediction

1.Project Overview

This project aims to predict second-hand housing prices using multiple machine learning models. By cleaning and selecting features from the dataset, and applying models such as Linear Regression, Ridge Regression, Random Forest, XGBoost, and LightGBM (with hyperparameter tuning), we compare their prediction accuracy. The models are evaluated using appropriate performance metrics and visualized to compare predicted and actual prices, leading to practical conclusions.

2.Data Loading and Preprocessing

The dataset used comes from second-hand housing transaction records in a specific city (sourced from: https://www.kaggle.com/c/house-prices-advanced-regression-techniques).

Preprocessing steps include:

Handling missing values (e.g., mean imputation, removing high-missing-rate features)

One-hot encoding for categorical variables

Outlier removal (e.g., filtering extreme values in area and unit price)

Feature standardization (normalizing certain numerical features)

The target variable is price (in RMB), with other columns serving as predictors.

3.Feature Engineering

Key feature engineering steps include:

Removing irrelevant features: such as ID or URL fields that provide no predictive value.

Feature derivation: generating new features like unit price = total price / area.

Encoding categorical variables: using one-hot encoding for fields like decoration status, floor level, region, etc.

Standardization/Normalization: applied to continuous features like area and building age.

4.Train-Test Split

The cleaned dataset is split into training and testing sets using an 80:20 ratio to assess generalization:

Training set: 80%

Test set: 20%

Random seed: 42 (for reproducibility)

5.Model Training and Tuning

Five different models were trained, with hyperparameter tuning for some:

(1)Linear Regression

Used as a baseline model without regularization.

Pros: Fast and interpretable.

Cons: Sensitive to outliers and prone to overfitting.

(2)Ridge Regression

Introduces L2 regularization to reduce feature collinearity effects.

Hyperparameter tuning:

alpha searched in [0.01, 0.1, 1, 10, 100] using GridSearchCV

Best alpha: 1.0

(3)Random Forest Regressor

An ensemble learning model suitable for nonlinear relationships.

Hyperparameters:

n\_estimators = [100, 200, 300]

max\_depth = [10, 20, 30]

Best: n\_estimators = 200, max\_depth = 20

(4)XGBoost Regressor

A boosting algorithm known for speed and performance, with feature importance analysis support.

Parameters:

learning\_rate = 0.1, n\_estimators = 200, max\_depth = 6

Shows excellent generalization on test data.

(5)LightGBM Regressor

Histogram-based gradient boosting model, more efficient than XGBoost.

Parameters:

learning\_rate = 0.1, n\_estimators = 200, max\_depth = 6

High prediction accuracy and fast runtime.

6. Model Evaluation Metrics and Results对所有模型分别在测试集上评估：

Model MAE RMSE

Linear Regression 20485.45 49257.88

Ridge Regression 20028.23 32649.01

Random Forest 17642.19 28770.67

XGBoost 17507.63 28180.96

LightGBM 17157.16 29372.53

Key Findings:

(1)Traditional linear models perform the worst:

Linear Regression had an RMSE of 49,257.88 RMB, indicating poor ability to capture complex price trends.

Ridge Regression improved upon this with regularization, but still lagged behind ensemble models.

(2)Ensemble models outperform linear models across the board:

Random Forest, XGBoost, and LightGBM showed significantly better MAE and RMSE values.

These models handle non-linear relationships and feature interactions effectively.

(3)XGBoost vs LightGBM:

Very close in performance.

XGBoost achieved slightly lower RMSE, while LightGBM had the lowest MAE, meaning it was more accurate for the majority of predictions.

XGBoost may reduce extreme prediction errors better.

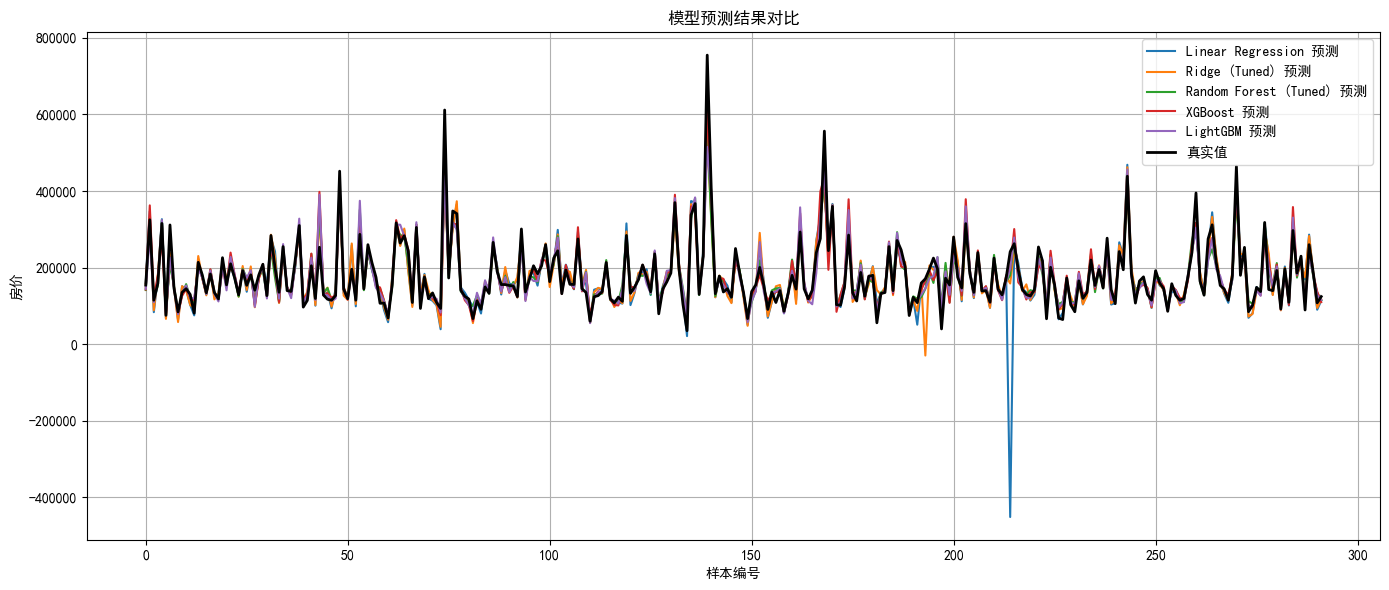
(4)LightGBM is the best overall performer:

Achieved the lowest MAE (17,157.16 RMB), indicating minimal average deviation.

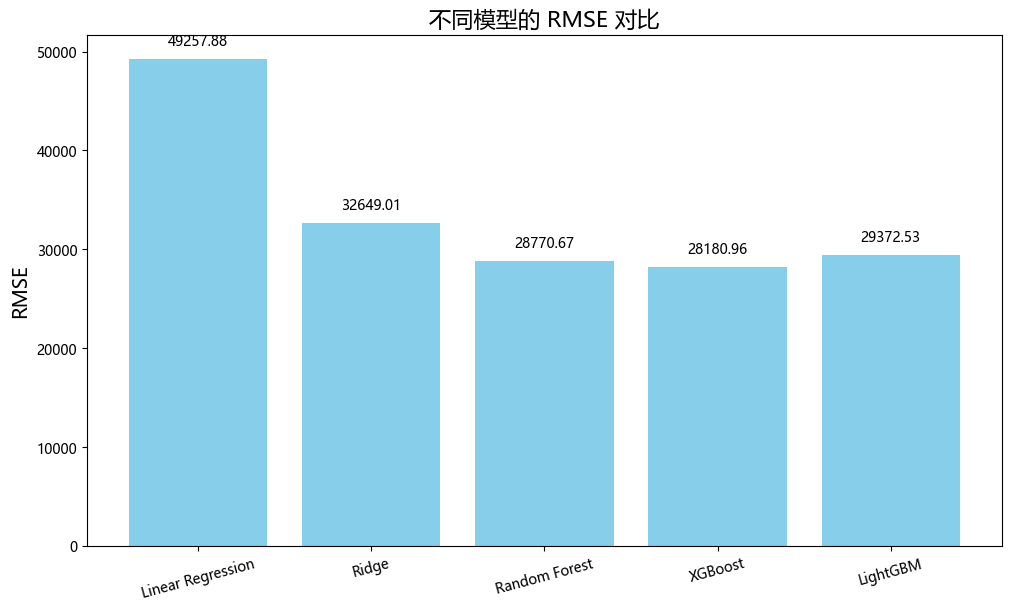
Though its RMSE was slightly higher than XGBoost, its prediction distribution was more stable and practical.

7.Visualization Results

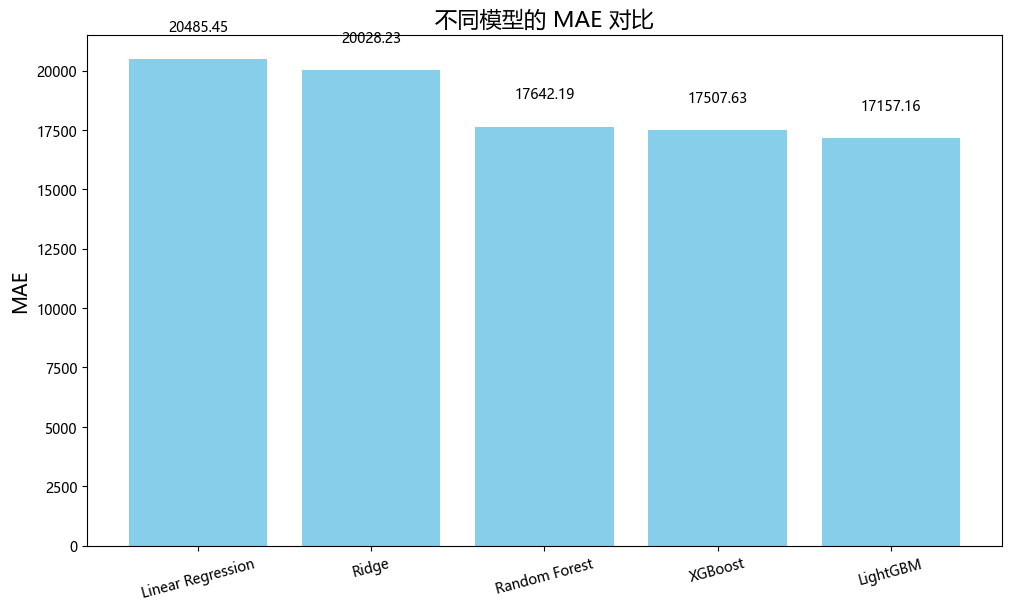
Comparison of model prediction results



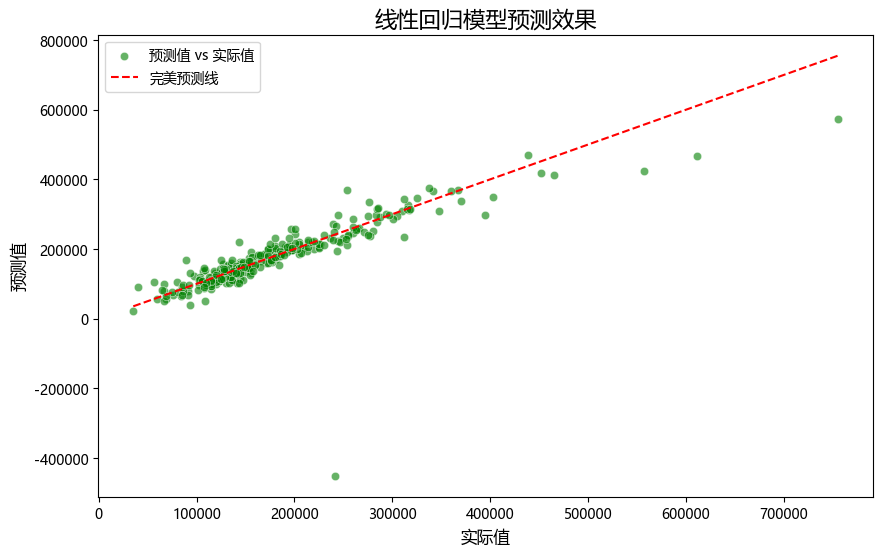
RMSE Comparison Across Models



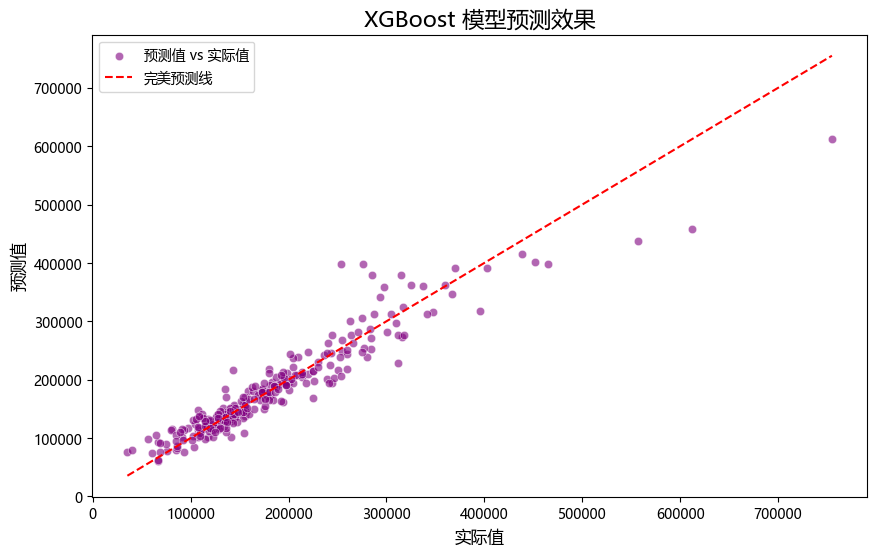
MAE Comparison Across Models



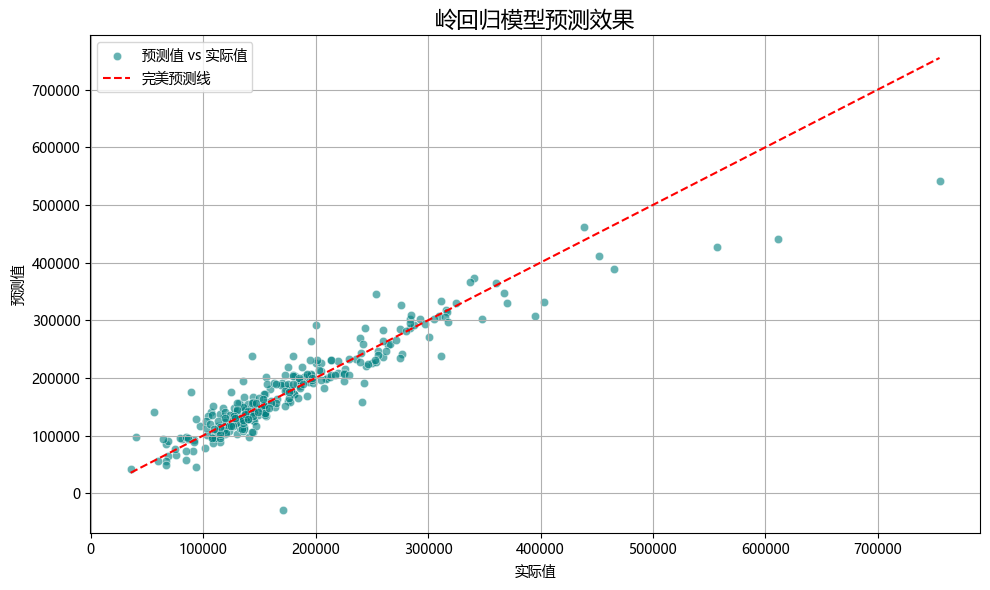
Linear Regression Model



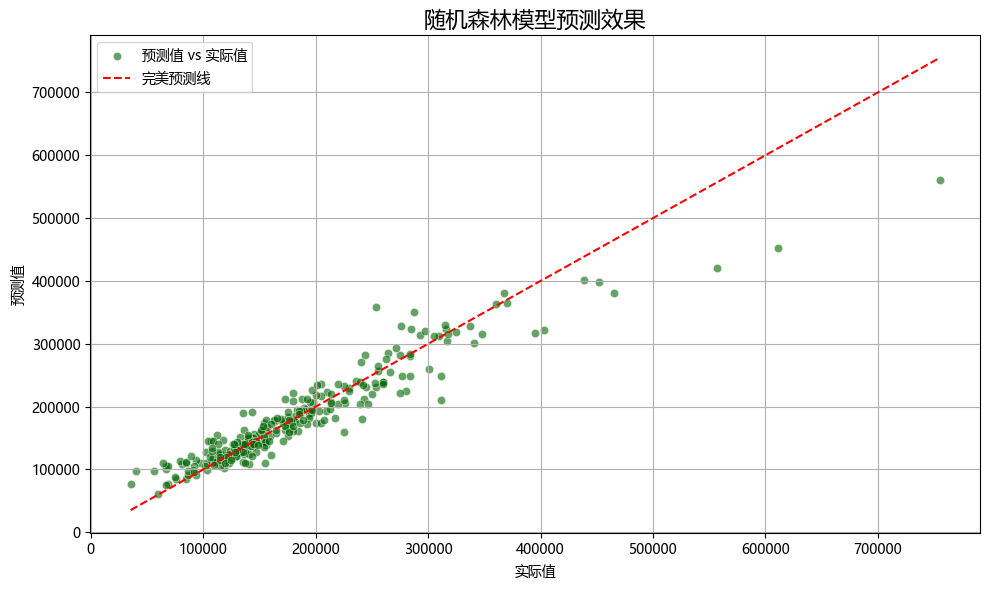
XGBoost Model



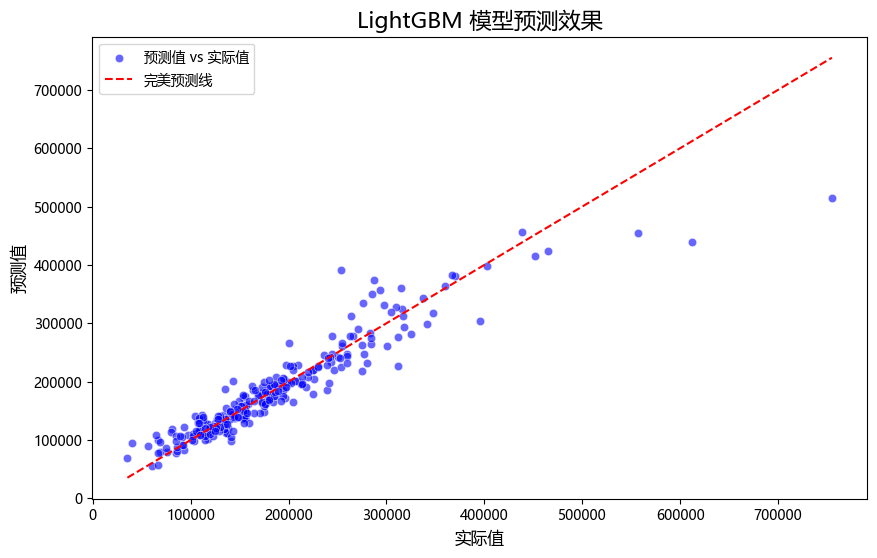
Ridge Regression Model



Random Forest Model



LightGBM Model



All Models Combined

